I still need to show you that this construction actually does what it is supposed to do.

E.g. Some string accepted by the PDA

$w = aabb$

Instantaneous descr.

$(q_0, aabb, z)$ $\qquad\qquad \boxed{q_0, z, q_f}$

$\vdash (q_1, abb, Az)$ $\qquad \rightarrow a \boxed{q_1, A, q_2} \boxed{q_2, z, q_f}$

$\vdash (q_1, bb, AAz)$ $\qquad \rightarrow aa \boxed{q_1, A, q_2} \boxed{q_2, A, q_2} \boxed{q_2, z, q_f}$

$\vdash (q_2, b, Az)$ $\qquad \rightarrow aab \boxed{q_2, A, q_2} \boxed{q_2, z, q_f}$

$\vdash (q_2, \cancel{b}\lambda, z)$ goes away here $\qquad \rightarrow aabb \boxed{q_2, z, q_f}$

$\vdash (q_f, \lambda, \lambda)$ $\qquad \rightarrow aabb$

This "proves" (to do it properly I should do it abstractly for every sequence of instantaneous descriptions) that every string accepted by the PDA is generated by the grammar.

I should now show you why every derivation in the grammar corresponds to a ~~sequ~~ valid sequence of instant. descr. for the PDA.

1. Make sure we use a leftmost derivation

2. The transitions we use at any given point is just the transition corresponding to the production (we don't have a choice - there is only one)

Deterministic PDAs

A PDA is deterministic if

① ~~We~~ do not have ~~two different~~ transitions

$$(q, \ell, s) \longrightarrow (q', t')$$
$$(q, \ell, s) \longrightarrow (q'', t'')$$

(where $q' \neq q''$ or $t' \neq t''$)

Intuitively - given the state we're at, the first letter remaining on the string and the top of the stack, we have at most one choice.

---

and - (we want λ-transitions so we can do stack stuff w/o eating string) but we don't want a choice btw a λ-transition and a string eating one.

② We do not have transitions

$$(q, \lambda, s) \longrightarrow (q', t')$$
$$(q, \ell, s) \longrightarrow (q'', t'') \quad \text{(where } \ell \neq \lambda\text{)}$$